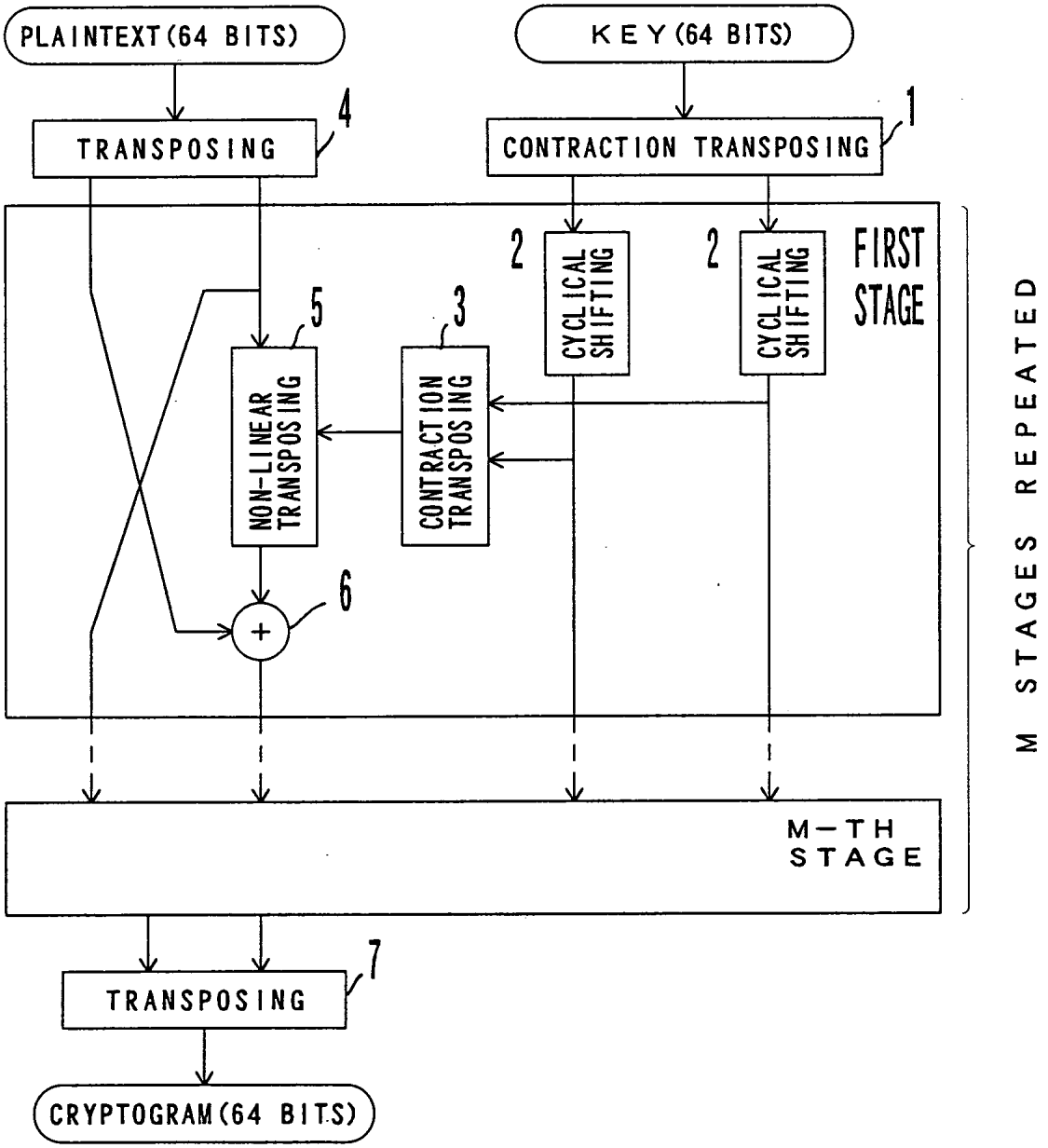
PLAINTEXT(64 BITS)
KEY(64 BITS)
TRANSPOSING
4
CONTRACTION TRANSPOSING
1
2
CYCLICAL SHIFTING
2
CYCLICAL SHIFTING
FIRST STAGE
5
NON-LINEAR TRANSPOSING
3
CONTRACTION TRANSPOSING
6
+
M STAGES REPEATED
M-TH STAGE
TRANSPOSING
7
CRYPTOGRAM(64 BITS)

FIG. 1

$n = p \cdot q$ (WHERE p AND q ARE PRIME NUMBERS)

$e \cdot d \equiv 1 \pmod{(p-1)(q-1)}$ ($e < n$, e AND $(p-1)(q-1)$ ARE RELATIVELY PRIME)

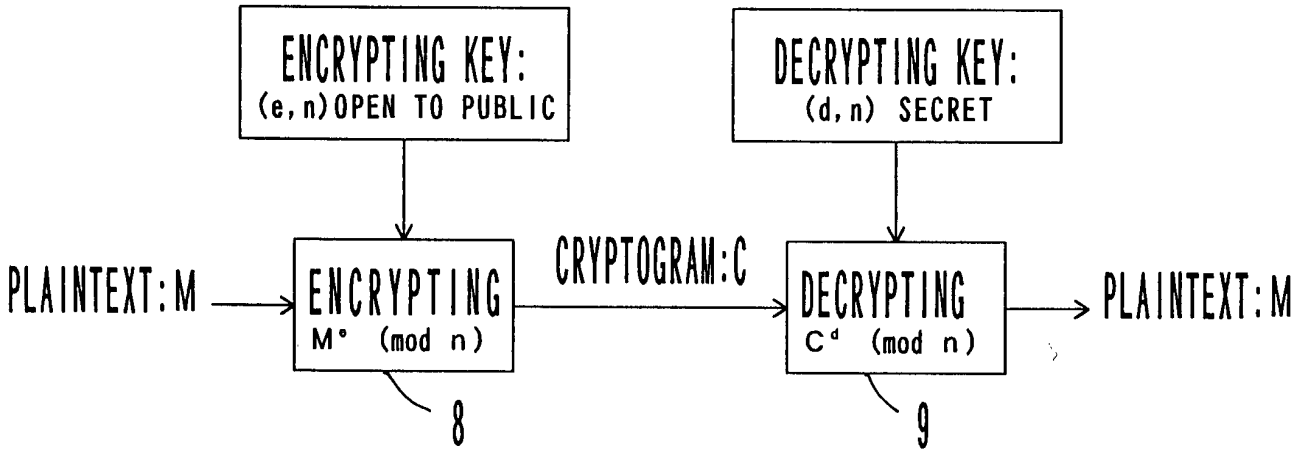

FIG. 2

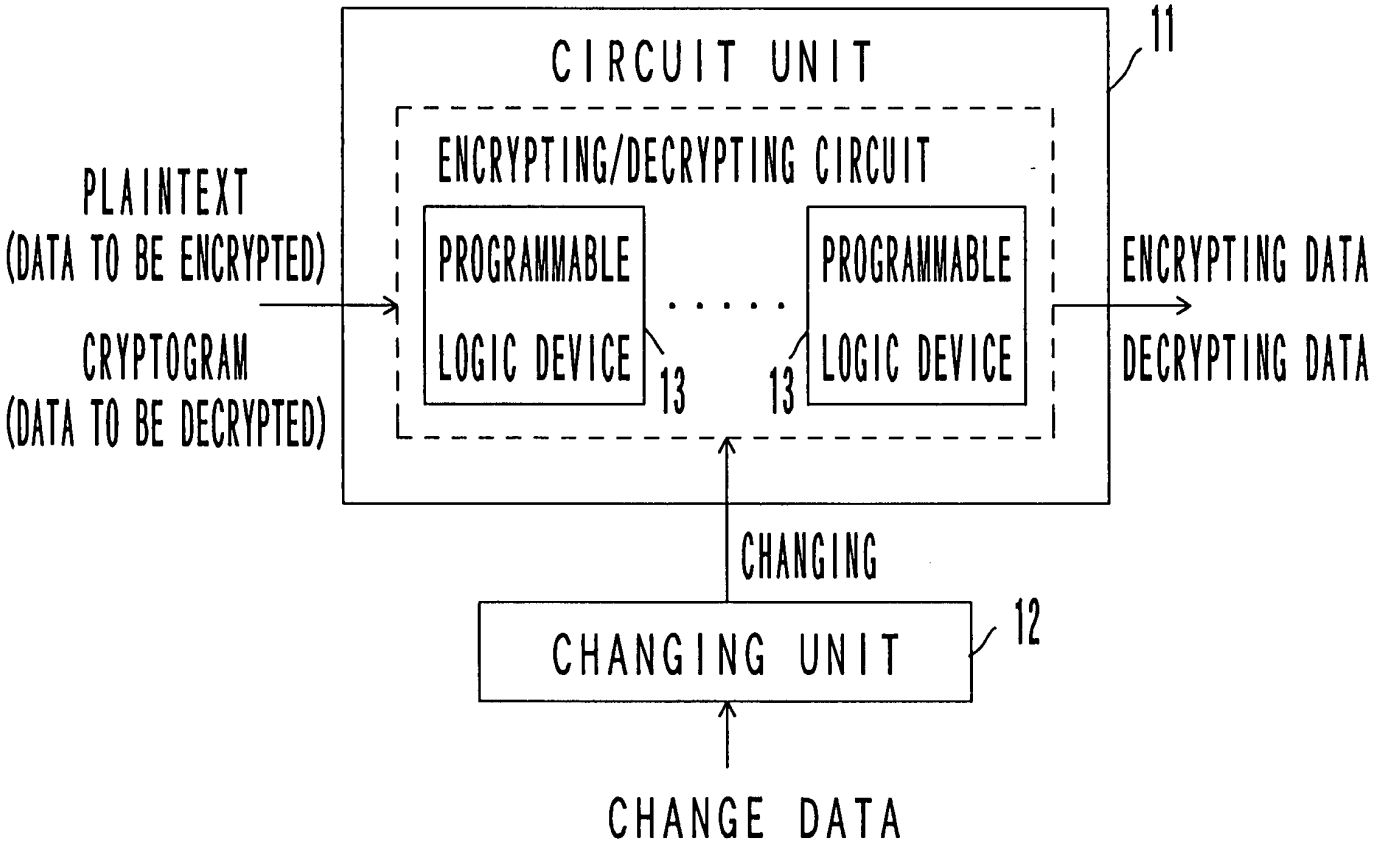

FIG. 3

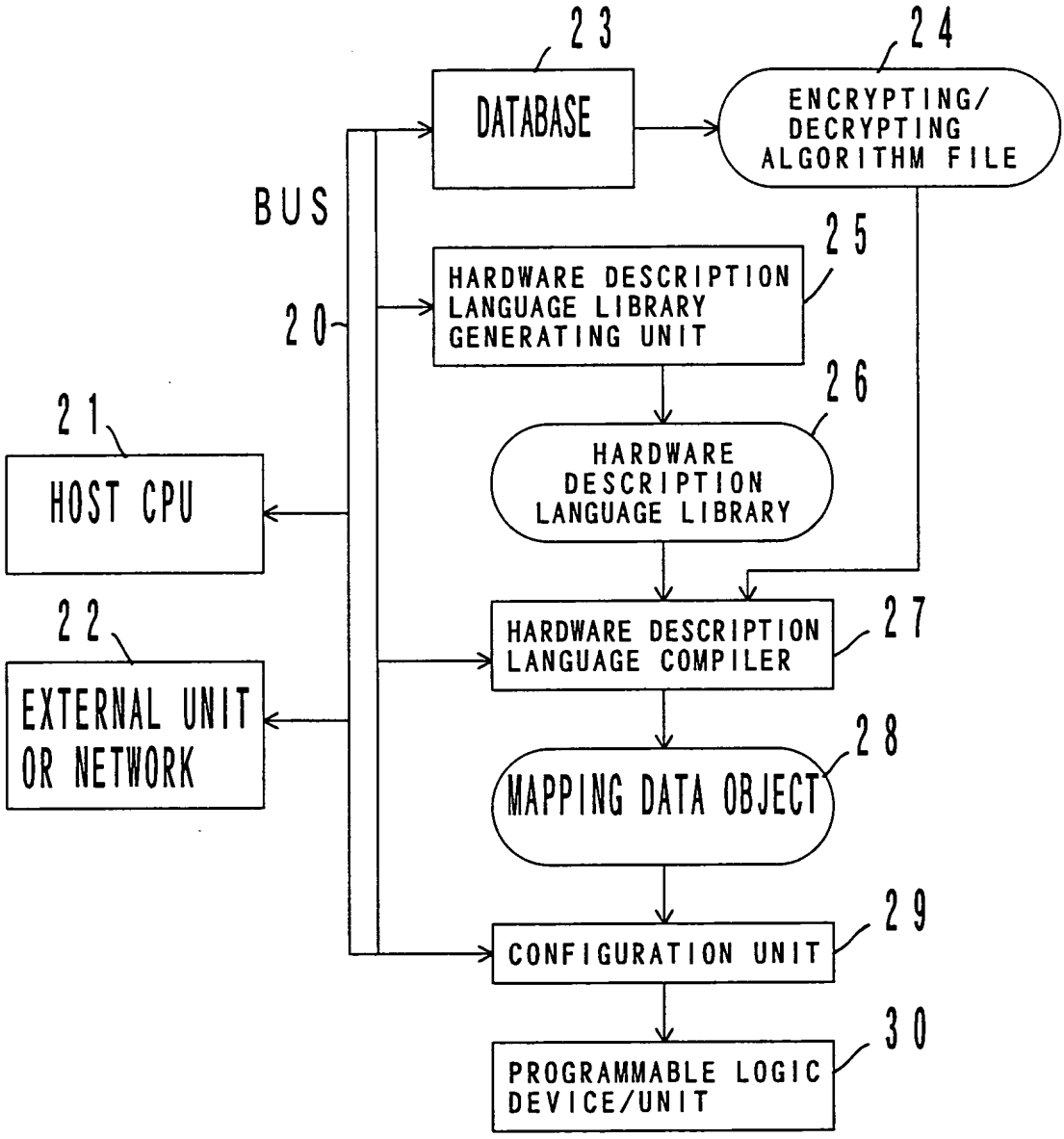
23
24
DATABASE
ENCRYPTING/
DECRYPTING
ALGORITHM FILE
BUS
25
HARDWARE DESCRIPTION
LANGUAGE LIBRARY
GENERATING UNIT
20
26
21
HOST CPU
HARDWARE
DESCRIPTION
LANGUAGE LIBRARY
22
27
HARDWARE DESCRIPTION
LANGUAGE COMPILER
EXTERNAL UNIT
OR NETWORK
28
MAPPING DATA OBJECT
29
CONFIGURATION UNIT
30
PROGRAMMABLE LOGIC
DEVICE/UNIT

FIG. 4

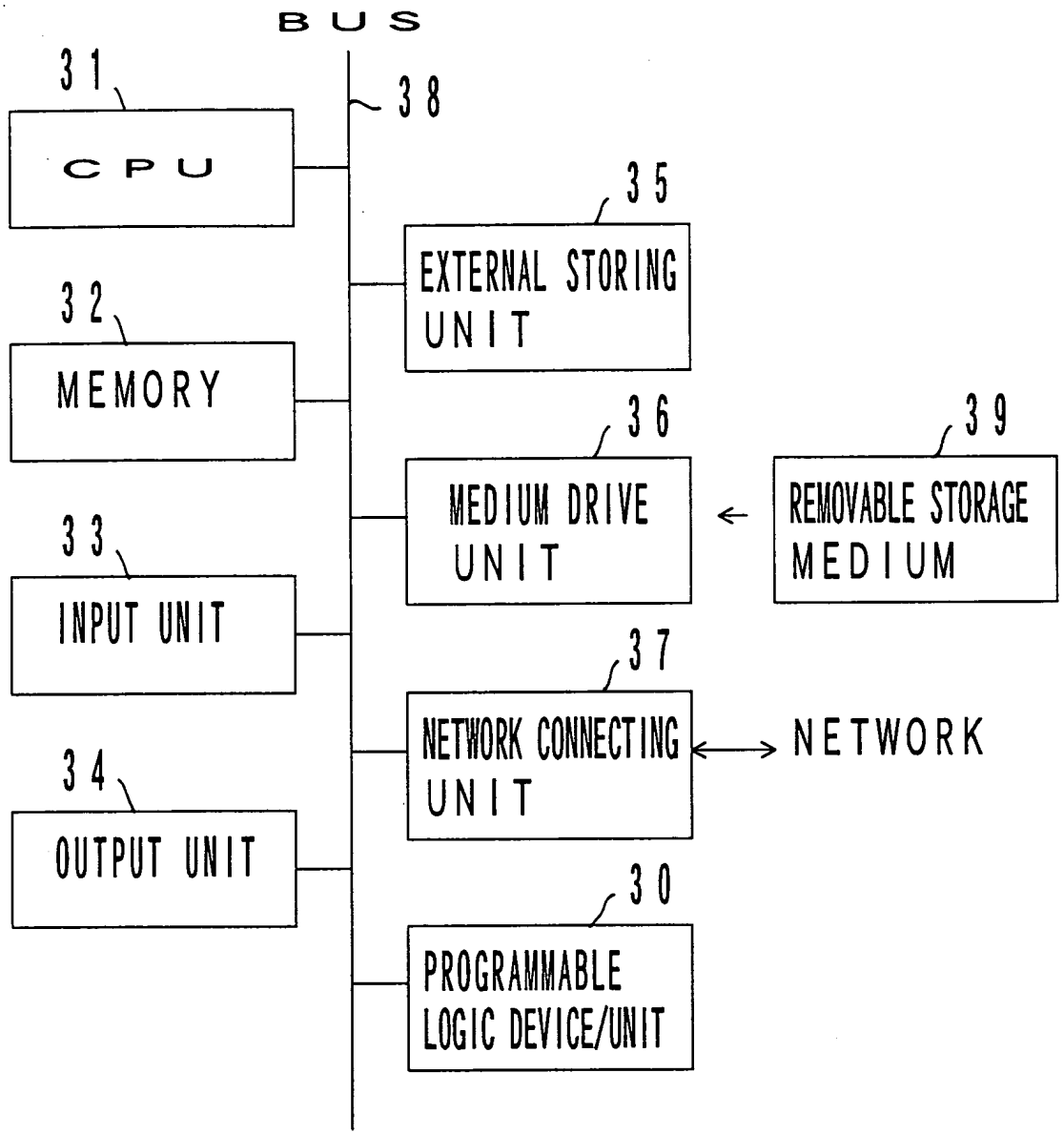
BUS
31
CPU
38
35
EXTERNAL STORING UNIT
32
MEMORY
36
39
MEDIUM DRIVE UNIT
REMOVABLE STORAGE MEDIUM
33
INPUT UNIT
37
NETWORK CONNECTING UNIT
NETWORK
34
OUTPUT UNIT
30
PROGRAMMABLE LOGIC DEVICE/UNIT

FIG. 5

```
module Bcount16(q, clk)
   output[15:0] q;
   input clk;
   reg[15:0] q;

   always@(posedge clk)
        q=q+ d1;
endmodule
```

FIG. 6

```
module top;
  reg clock, reset, start,  end;
L1— wire [b1:0] M, C;   <-15
L2— wire [b2:0] e;      <- 7
L3— wire [b3:0] n;      <-63

rsaEnc     enc1(M, C, e, n, clock, reset, start, end);

endmodule

module rsaEnc(M, C, e, n, clk, res, st, ed);
   input [b1:0] M;   <-15
   input [b2:0] e;   <- 7
   input [b3:0] n;   <-63
   input  clk, res, st;
   output [b1:0] C; <-15
   output ed;
  {
    integer i;
    always@(posedge clk)
    if (res == 1`b1)
       C = 16`d0;
    else if (st == 1`b1)
           C = 1`b1;
           for (i=0;i<e;i++) {
                C=(M*C)%n;
           }
           ed=1`b1;

  }
endmodule
```

FIG. 7

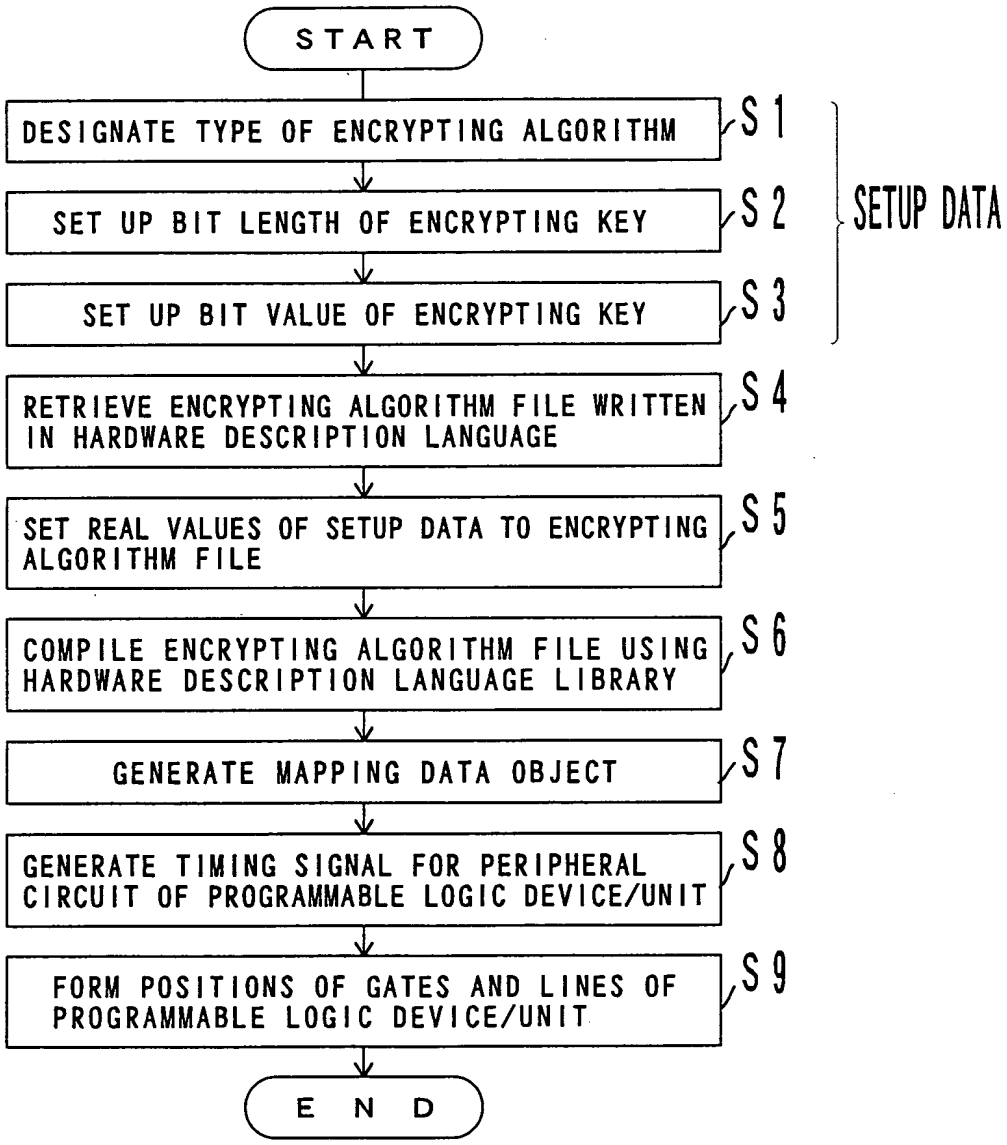
START
DESIGNATE TYPE OF ENCRYPTING ALGORITHM
S1
SET UP BIT LENGTH OF ENCRYPTING KEY
S2
SETUP DATA
SET UP BIT VALUE OF ENCRYPTING KEY
S3
RETRIEVE ENCRYPTING ALGORITHM FILE WRITTEN IN HARDWARE DESCRIPTION LANGUAGE
S4
SET REAL VALUES OF SETUP DATA TO ENCRYPTING ALGORITHM FILE
S5
COMPILE ENCRYPTING ALGORITHM FILE USING HARDWARE DESCRIPTION LANGUAGE LIBRARY
S6
GENERATE MAPPING DATA OBJECT
S7
GENERATE TIMING SIGNAL FOR PERIPHERAL CIRCUIT OF PROGRAMMABLE LOGIC DEVICE/UNIT
S8
FORM POSITIONS OF GATES AND LINES OF PROGRAMMABLE LOGIC DEVICE/UNIT
S9
E N D

FIG. 8

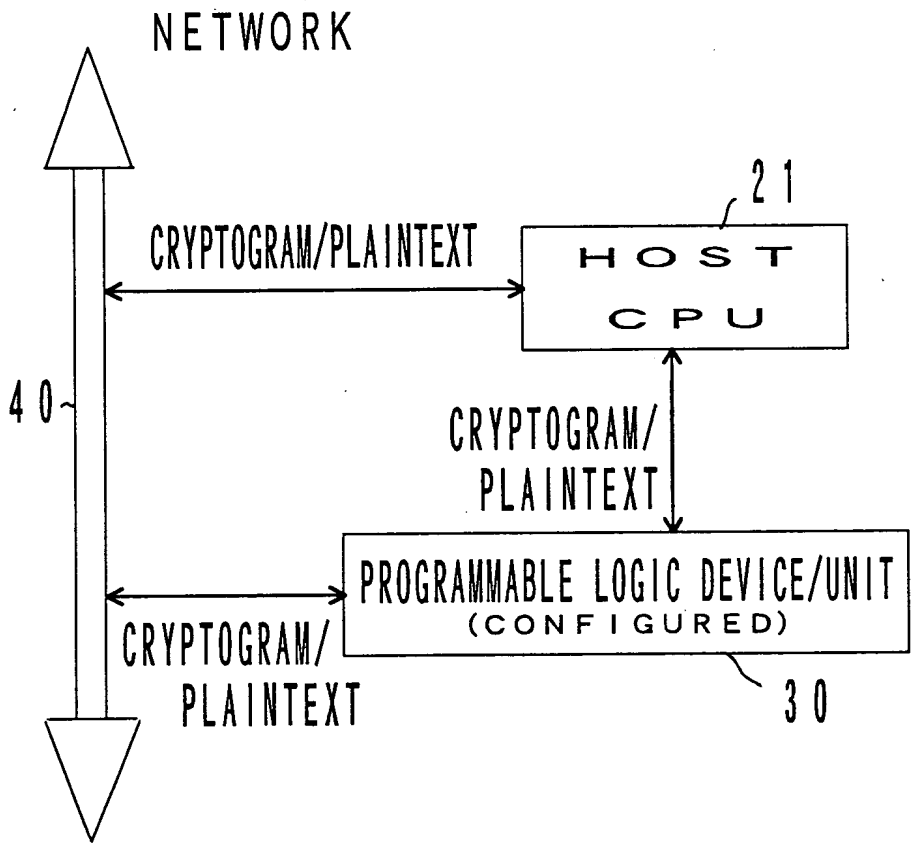
NETWORK
21
CRYPTOGRAM/PLAINTEXT
HOST
CPU
40
CRYPTOGRAM/
PLAINTEXT
PROGRAMMABLE LOGIC DEVICE/UNIT
(CONFIGURED)
CRYPTOGRAM/
PLAINTEXT
30

FIG. 9

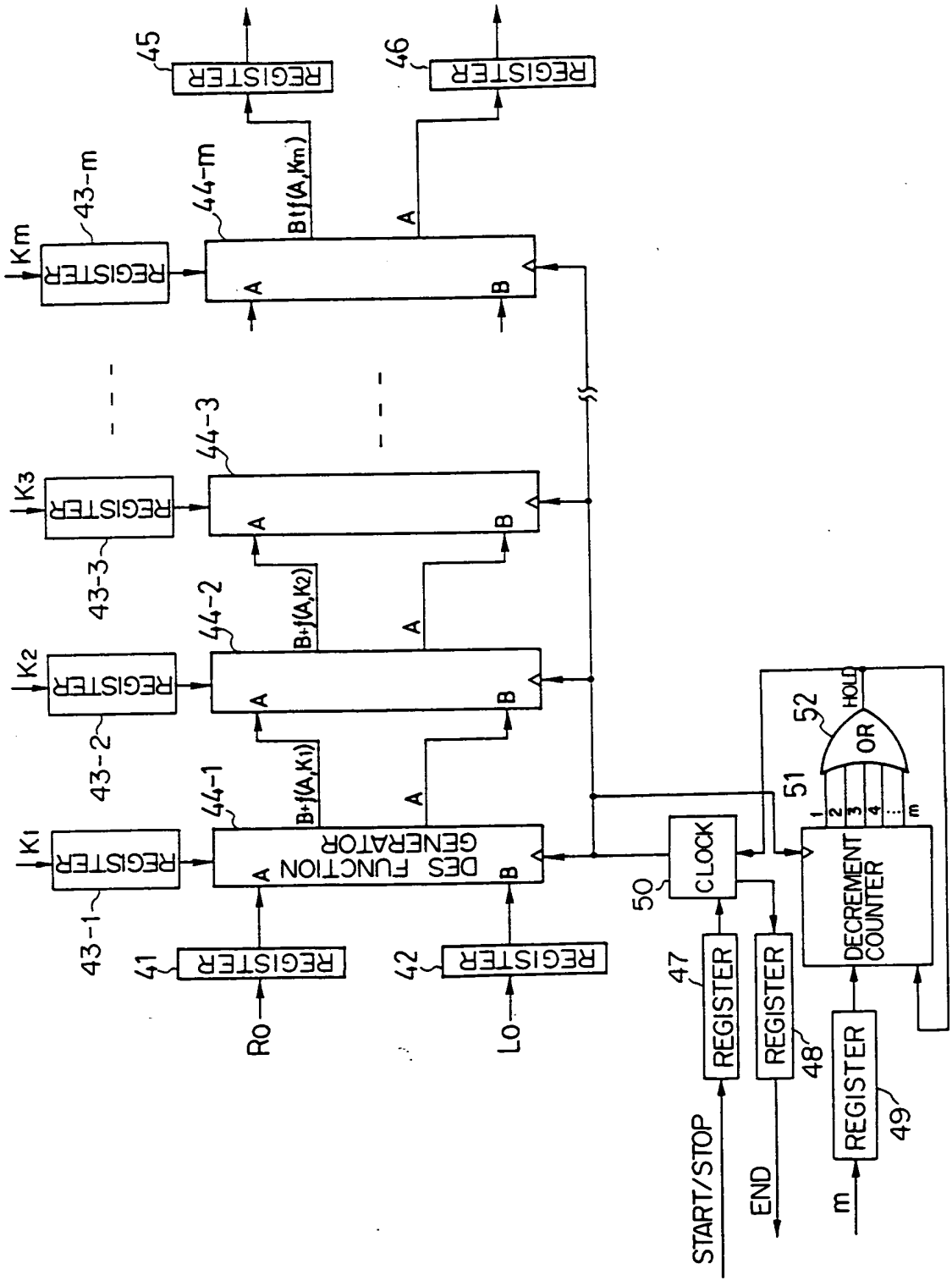
45
REGISTER
46
REGISTER
43-m
44-m
Km
REGISTER
B1f(A,Km)
A
B
A
B
43-3
44-3
K3
REGISTER
B+f(A,K2)
A
B
A
B
43-2
44-2
K2
REGISTER
B+f(A,K1)
A
B
A
B
43-1
44-1
K1
REGISTER
DES FUNCTION GENERATOR
A
B
41
REGISTER
42
REGISTER
R0
L0
50
CLOCK
51
52
OR
HOLD
1
2
3
4
...
m
DECREMENT COUNTER
47
REGISTER
48
REGISTER
49
REGISTER
START/STOP
END
m

FIG. 10

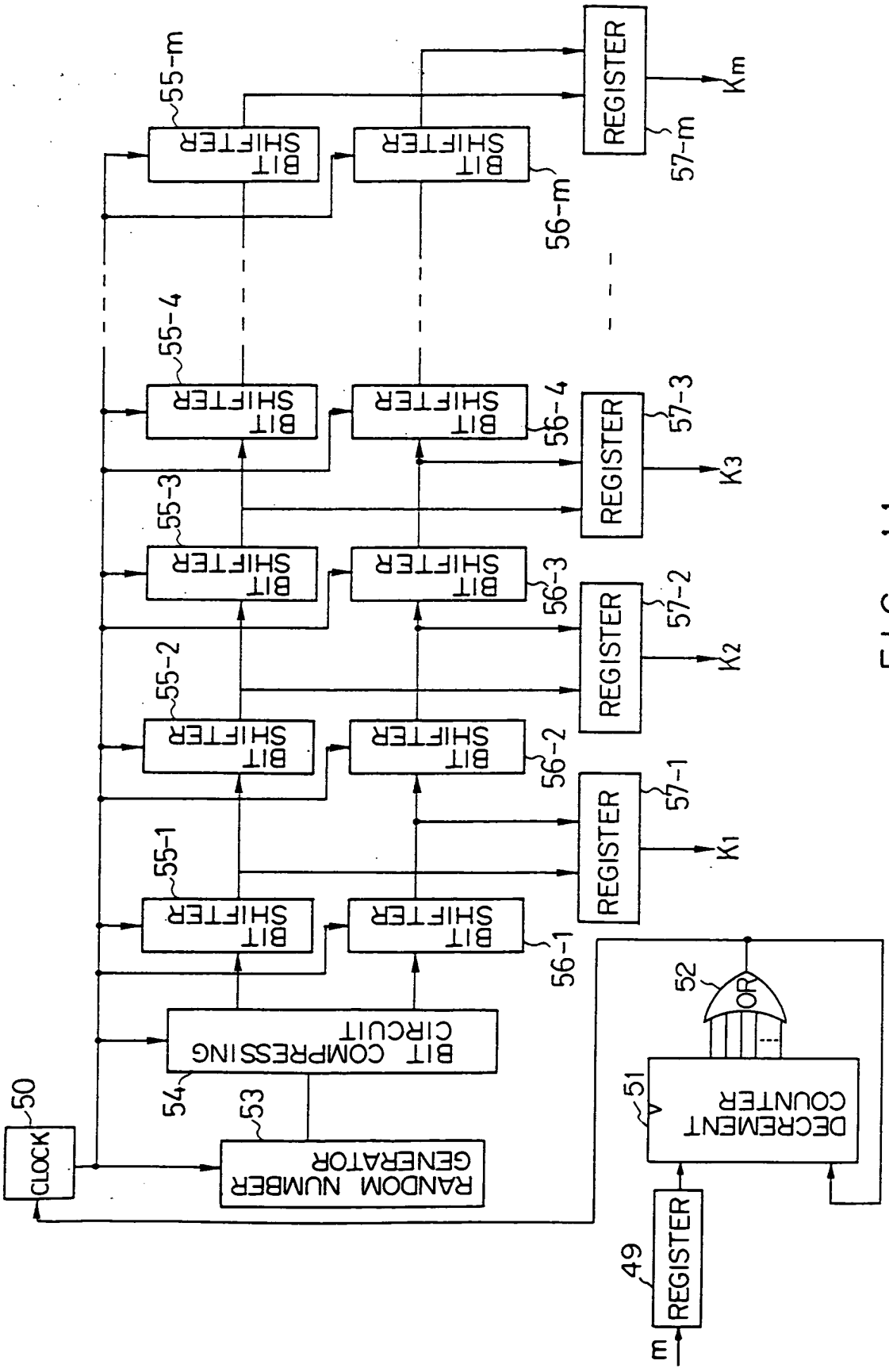
m
REGISTER
49
DECREMENT COUNTER
51
OR
52
CLOCK
50
RANDOM NUMBER GENERATOR
53
BIT COMPRESSING CIRCUIT
54
BIT SHIFTER
55-1
55-2
55-3
55-4
55-m
56-1
56-2
56-3
56-4
56-m
REGISTER
57-1
57-2
57-3
57-m
K1
K2
K3
Km

FIG. 11

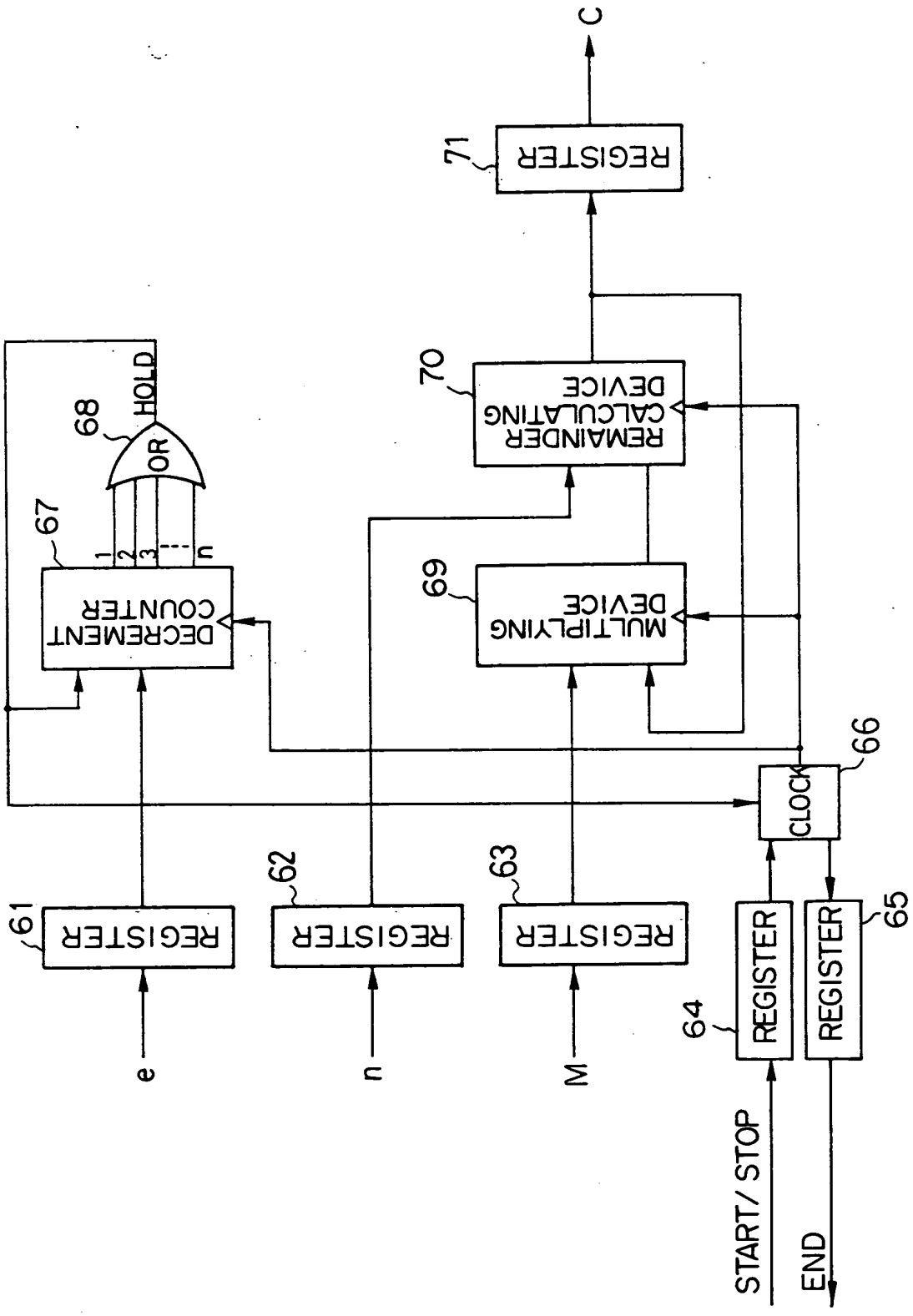
C
71
REGISTER
70
REMAINDER CALCULATING DEVICE
68
HOLD
OR
67
1
2
3
n
DECREMENT COUNTER
69
MULTIPLYING DEVICE
66
CLOCK
61
62
63
REGISTER
REGISTER
REGISTER
e
n
M
64
65
REGISTER
REGISTER
START/ STOP
END

FIG. 12

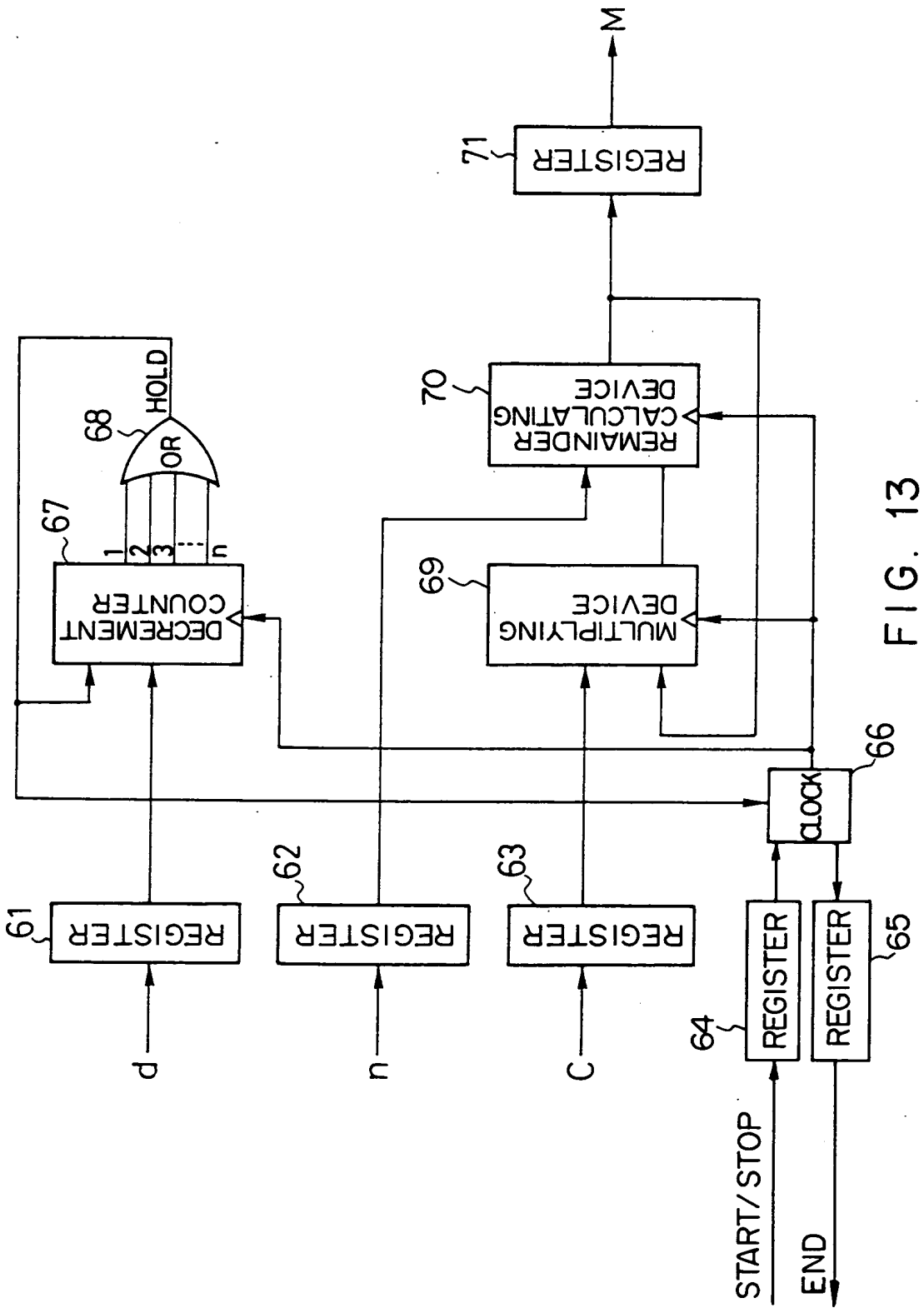

FIG. 13

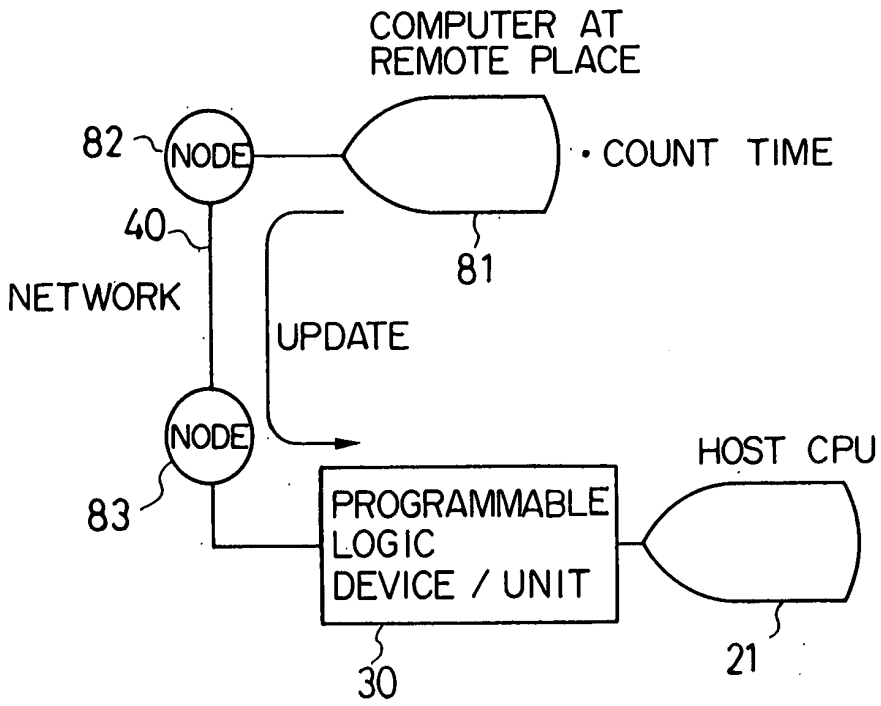

FIG. 14

08/814409

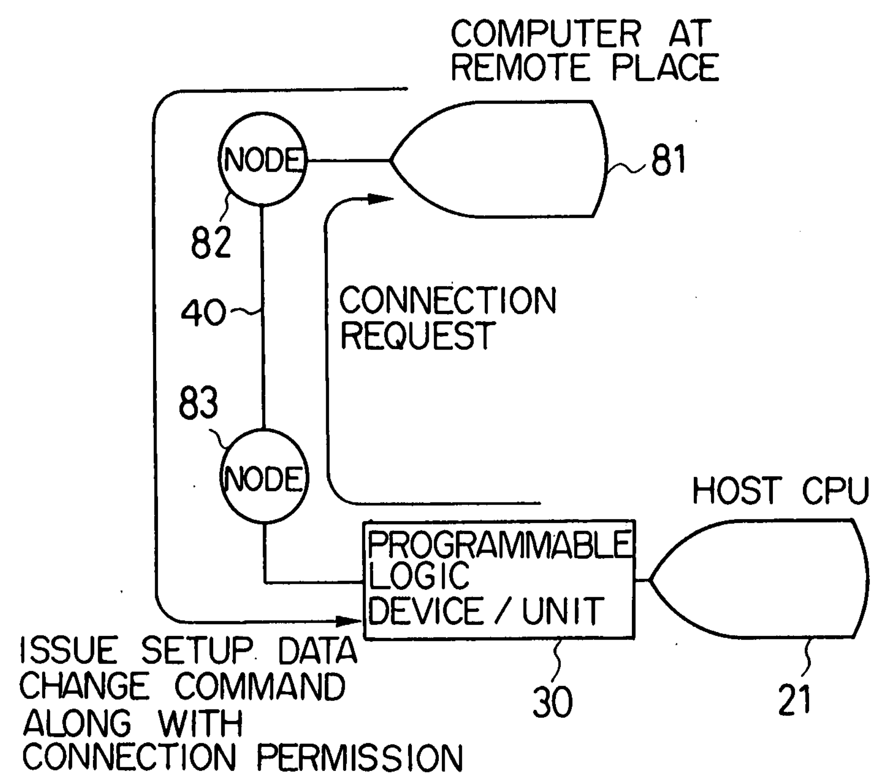

FIG. 15

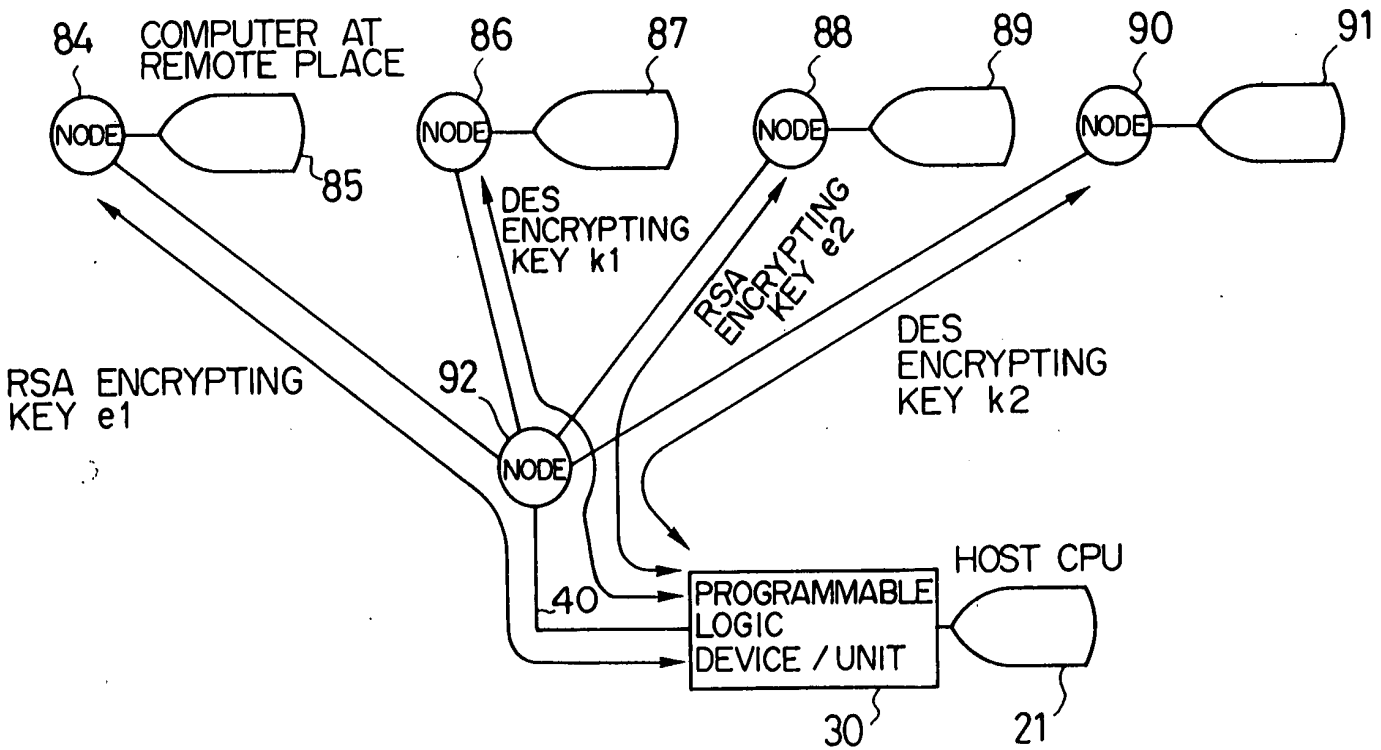

FIG. 16

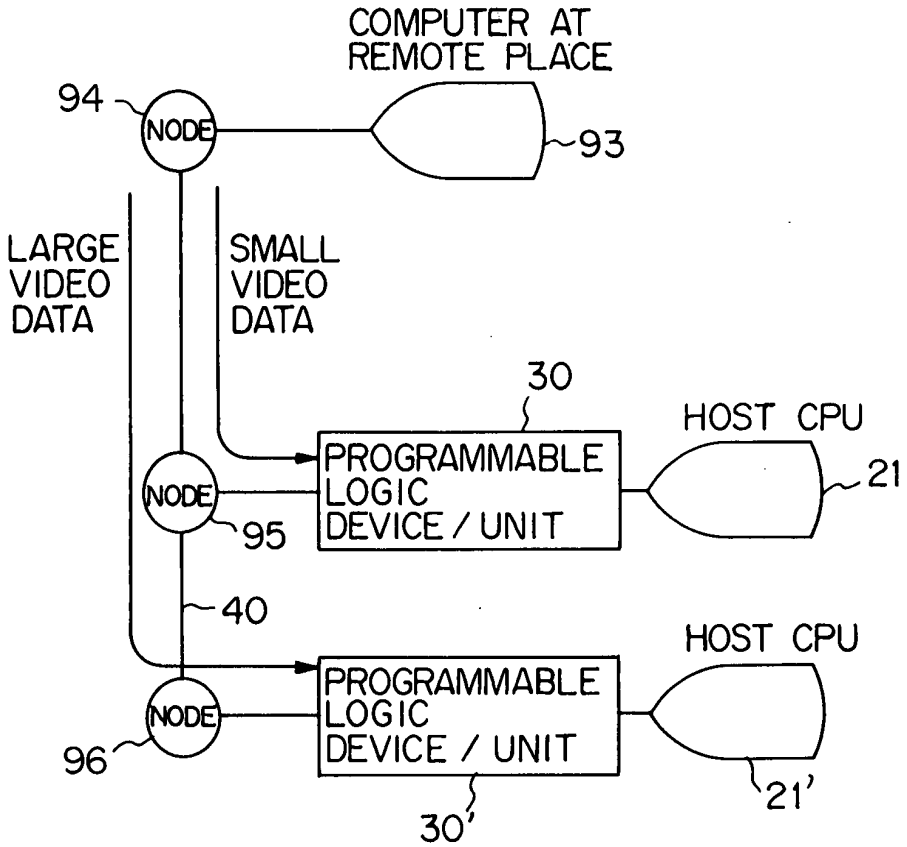

FIG. 17